A Survey of User Behavior in VoD Service and Bandwidth-Saving Multicast Streaming Schemes

Joonho Choi, Abu (Sayeem) Reaz, and Biswanath Mukherjee

***Abstract*—As broadband networks using Fiber-to-the-x (FTTx) technologies are being increasingly deployed in access networks, video service, especially VoD (Video-on-Demand), is becoming more attractive to deploy. To provide efficient VoD service, a cost-effective service model is very important and a lot of research has been conducted over the past decade. This paper reviews the existing literature on this topic, focusing on the user behavior in VoD services and bandwidth-saving multicast streaming schemes, which are the most important aspects of VoD service. First, we review the user behavior in VoD such as video popularity, daily access pattern, and interactive VCR (Videocassette Recorder) properties from recent data. Each video title's rental frequency, i.e., video popularity, follows the Zipf distribution, and this popularity can change with time or by service provider's recommendation of videos. This overall request frequency for each video constitutes a specific pattern throughout the day and has a similar pattern every day. Second, we review the bandwidth-saving streaming schemes such as broadcasting, batching, patching, and merging, which use multicast streaming technologies and user buffer memory. We review the mechanism of each multicast streaming technology and compare their differences. We also review the recent trends on multicast streaming technologies, which is summarized as hybrid architecture which combines several multicast streaming technologies to obtain better performance. Next, we review how these multicast streaming technologies implement interactive VCR functions. We classify the VCR interactivity into discontinuous and continuous VCR actions and examine the principles for VCR support in multicast streaming schemes: caching some video data for discontinuous VCR support and allocating contingency channels for continuous VCR support. We review mechanisms of VCR support for different multicast streaming schemes. Through this survey, we provide an in-depth understanding of VoD service deployment.**

***Index Terms*—FTTx, User Behavior, Video-on-Demand (VoD), VCR, Zipf, Multicast, Streaming.**

I. Introduction

IN RECENT years, Fiber-to-the-x (FTTx) technologies are being increasingly deployed in access networks, and substituting xDSL Digital Subscriber line, which has been a dominant technology represented by ADSL and VDSL for broadband access service for the past few years. Due to the rapid development of semi-conductor memory and digital TV technologies, the customer-side equipment, which receives and displays broadband service, is also diversified - from PC to TV equipped with set-top box and PVR (Personal Video Recorder) which can store a large amount of video data (e.g., 120 Gbytes) [1]. Therefore, traditional service paradigms, which focused on Internet access, are evolving and expanding to true triple-play service represented by voice, Internet (data), and video.

Manuscript received 4 June 2009; revised 10 November 2010 and 30 November 2010.

The authors are with University of California, Davis, USA (e-mail: {jhch,asreaz,bmukherjee}@ucdavis.edu).

Digital Object Identifier 10.1109/SURV.2011.030811.00051

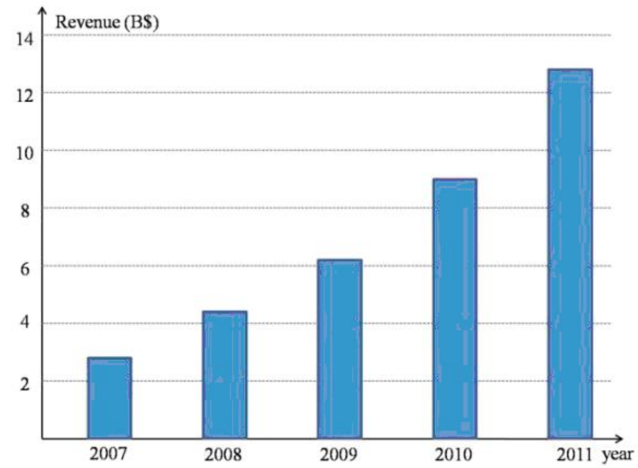


Fig. 1. World-wide VoD service revenue forecast (2007 - 2011) [3].

Video in triple-play service was not easy to provide in the past due to its large bandwidth requirement, but now it is becoming possible. Especially, Video-on-Demand (VoD) service, which provides video titles to user homes based on user requests by online access without going to video rental shop, is regarded as a very promising service for revenue source among various video services. Therefore, service providers believe that VoD service will become a competitive differentiator in establishing new subscribers, and they are developing cost-effective solutions to provide this service. Several research institutes analyzed and forecasted the bright future of the VoD service based on the technological developments and prospects. Reference [2] reports that the number of homes, which are connected with broadband and are ready for on-demand contents, was 308 million worldwide in 2008. Reference [3] forecasts that VoD market will grow from $2.7 billion in 2007 to $12.7 billion in 2011 in worldwide revenue as shown in Fig. 1. Reference [2] forecasts that worldwide annual revenue for movie and TV shows from rental transaction on pay-TV and online platforms will grow from $2.2 billion in 2008 to $3.7 billion in 2012.

VoD service is inherently a personalized service from the characteristic of one-to-one interaction, so it requires a dedicated stream to serve each user request. However, to provide a separate (high-bandwidth) video stream for each user request is not cost-effective with limited network bandwidth. Therefore, over the past decade, quite a lot of research has been conducted to provide VoD service economically.

1553-877X/12/$25.00 © 2012 IEEE

Two popular distribution models for VoD services are client-server and point-to-point (P2P) models. In this survey, we mostly focus on client-server models, i.e., from the perspective of a VoD service provider. Two popular research directions in client-server-based VoD services are: i) analysis of user behaviors such as video access patterns and VCR-like interactions, and ii) minimizing the usage of bandwidth using multicast streams and how these schemes implement interactive VCR operations. Together, user access patterns, VCR interaction pattern while accessing, and multicast streams to provide bandwidth savings for these interactions can ensure efficient VoD services with VCR functionalities.

Literature on the client-server model has appeared over the past decade. Though some of the references are dated, their contributions are important and essential for understanding of VoD. Hence, we present a survey of research works on both aspects of client-server distribution model of VoD to provide an understanding of the service. First, we discuss the user behavior analysis in the literature such as video access patterns and VCR-like interactions. A user behavior model helps to decide the overall network traffic and to develop a better strategy to provide VoD service so that the service provider can maximize the given resources such as network bandwidth and user storage. Then, we survey the most well-known multicast streaming schemes such as broadcasting [4], [5], [6], [7], batching [8], [9], [10], [11], patching [12], [13], [14], and stream merging [14], [15], [16], and discuss how interactive VCR operations are implemented in multicast streaming schemes. Unlike the traditional concept for VoD service, which provides a dedicated unicast stream to each request, these schemes use a multicast stream and exploit the user buffer to store some of the video data in user equipment. Thus, these schemes can reduce the required bandwidth for the service and optimize the usage of network bandwidth while providing almost the same quality of service to users.

The rest of the paper is organized as follows. First, we mention some of the important additional literature on VoD not covered in our survey in Section II. As the contributions in VoD are vast and there are several aspects of research on VoD, we recognize the other important dimensions in VoD research. In Section III, we examine the user behavior in VoD service such as video access pattern, video popularity, and interactive VCR action characteristics. Since most previous research on user behavior has been limited to the usage of educational videos from universities, we focus on the recent practical data obtained from the real environment of several service providers. In Section IV, we review the popular bandwidth-saving multicast streaming schemes to understand the concepts and mechanisms of these schemes, and compare their differences. We also review the recent trends on multicast streaming schemes, which are summarized as hybrid architecture which combines several multicast streaming technologies to obtain better performance. We provide a detailed comparison of different multicast schemes and their playback style and bandwidth and buffer requirements. In Section V, we first introduce the principles for VCR support in multicast schemes, and then review the interactive VCR support mechanisms in multicast streaming schemes, which include discontinuous VCR support by caching some of video data in user buffer and continuous VCR support by allocating contingency channels. We also review VCR implementation mechanisms in different multicast streaming schemes. Finally, we compare and contrast the implementations of interactive VCR functionality for different multicast schemes.

II. Additional Literature on VoD

The literature on VoD is rich and there are different possible research dimensions. Each of these research areas is important and challenging. However, as we mostly focus on client-server-based VoD services and their aspects, we do not provide in-depth discussions for other research dimensions.

More *recent* works on VoD focus on specialized broadcasting [17], [18], [19], patching [20], [21], [22], and batching [23] schemes for VoD services over wireless and mobile networks. A survey on multimedia streaming on wireless sensor networks is presented in [24]. However, we do not elaborate on these schemes as they are out of the scope of this survey.

Note that P2P-based VoD services are becoming popular, and there are a number of works in the literature that focus different aspects of P2P VoD services such as optimization and evaluation [25], caching and storage [26], [27], dissemination [28], searching [29], measurement [30], architecture [31], patching [32], VCR/DVD functionality [33], [34], robustness [35], and mobility [36], [20]. However, we do not provide a in-depth description on issues regarding P2P as they are out of the scope of our survey.

Another related work on VoD is workload analysis of streaming media [37], [38], [39]. They are mostly measurement studies where user data is extensively studied for analysis of the workload of the streaming media over the Internet. Importance of peer assistance to reduce server bandwidth is studied in [40]. Here, empirical data is analyzed for VoD services over the Internet, and it is shown that P2P-based VoD architecture can improve server bandwidth usage. However, the focus is not on user behavior specific to video accessing, rather on how the streaming media impacts the network and its parameters and profitability. Hence, we do not provide a detailed discussion on these topics.

There are several other notable works on video streaming in the literature. Security issues such as authentication of multicast source over unreliable video streaming protocol is discussed in [41]. A survey on important congestion control schemes for multicast video streaming over the Internet is presented in [42]. A tutorial on video-trace based evaluation of streaming video over network is presented in [43] where authors discuss statistical properties of video traces and their impact on the transport of video over networks.

III. User Behavior in VoD Services

For the service provider to provide VoD service efficiently, it is very important to understand the user behavior pattern, and a lot of research has been conducted to analyze it. Most previous efforts tried to analyze user behavior by using educational videos from lectures in university environment [44], [45], [46], which is different from real VoD environment, since most of university videos are relatively short and users use more

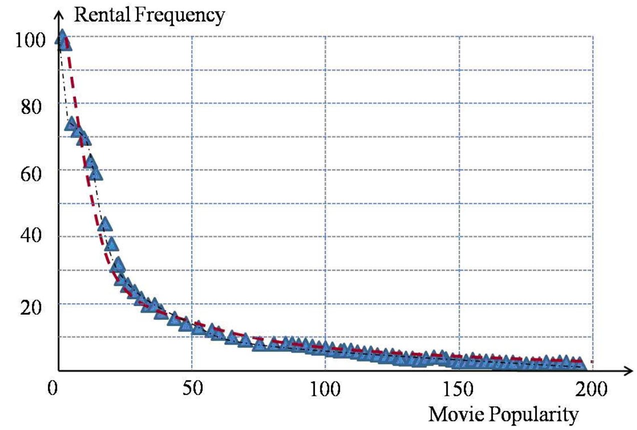


Fig. 2. Video rental frequency and Zipf distribution [50].

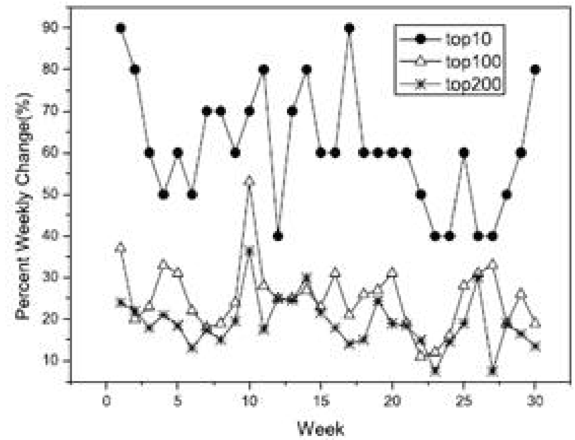


Fig. 3. User interest change over a month [48].

interactive functions to study the lectures. Recent research has tried to capture the practical environment and analyzed user behavior more realistically [47], [48], [49], [50], [51], [52]; e.g., Ref. [49] compared the client characteristics of educational videos, entertainment videos, and entertainment audios using the logs of eTech media server delivering educational contents at major US universities and the logs of two Latin America service and content providers. Reference [47] analyzed user behavior from the 4-year log data of www.lne.es (La Nueva Espaa Digital, LNE) VoD service in Spain. Reference [48] presents detailed analysis results from the seven-month log data of VoD system deployed by China Telecom. From the different characteristics of stored data and user access patterns, these results do not exactly coincide with each other in some behavior patterns, but the general user behaviors such as video access frequency and some of the user's interactive actions can be observed. In our study, we mainly focus on user behavior on entertainment videos such as movies, dramas etc., which is the service provider's main concern. However, we present some aspects of user behavior from educational videos as well for comparison purposes.

Here, we first present video access patterns of VoD services, and discuss the VCR interaction patterns for such accesses. They provide insight on how to predict and estimate the behavior users on a per-video, per-interaction basis.

A. Video Access Pattern

When the service provider provides VoD service, each video title has different popularity, and the popularity of a video correlates with the request rate for the video. The popularity of a video may decrease with time, since user interest for a video decreases after watching it. The popularity of a video may also change due to external factors such as introduction of new videos and recommendation of videos [48]. Users may have different preferences for each video and have diverse preferable time to watch a video. However, the overall request patterns from all clients follow a uniform curve and constitute daily and weekly access patterns, which are very important parameters for the service provider to consider for their network design.

1) **Video Popularity:** It is well known that video popularity follows the Zipf distribution, which means that, among v video titles, the popularity of video i has access frequency (probability) of C/i, where C is a normalization constant satisfying the equation ($\sum_{i=1}^{v} \frac{C}{i^{\alpha}} = 1$) and α is the distribution parameter. Figure 2 shows the relationship between real rental frequency (triangle in Fig. 2) and Zipf distribution curve (dotted line in Fig. 2). Some research papers such as [44] and [53] argue that the Zipf curve does not fit for video files, which has fetch-at-most-once property, but fits better for web objects by showing log-log plot. However, many recent practical results such as [47], [48], [49], which are experiments in real environment, show that video files follow the Zipf distribution. References [44], [45], [49] show that access rates for audio and educational videos can be modeled with the concatenation of two Zipf distributions.
2) **Change of User Interest for a Video:** Video popularity, i.e., request frequency, follows the Zipf distribution, but video popularity may change with time due to several reasons. Therefore, it is important to consider the changes of user interest for a video to determine its popularity, since user interest for a video eventually decreases with time. Reference [54] shows that most accesses for a file occur shortly after the file is introduced, and after a while, the access rate gradually decreases over time. User request rate for a video may also change due to external factors such as introduction of new video titles or recommendation of specific videos by the service provider.

Reference [48] shows how user interest changes over a day, over a week, and over weeks from the real log data; in a day, user interest does not change a lot except for early in the morning when new titles are usually introduced. For a long period of time such as over weeks, as shown in Fig. 3, user interest for top-10 videos significantly changes and percent weekly change is very high compared to top-100 or top-200 videos.

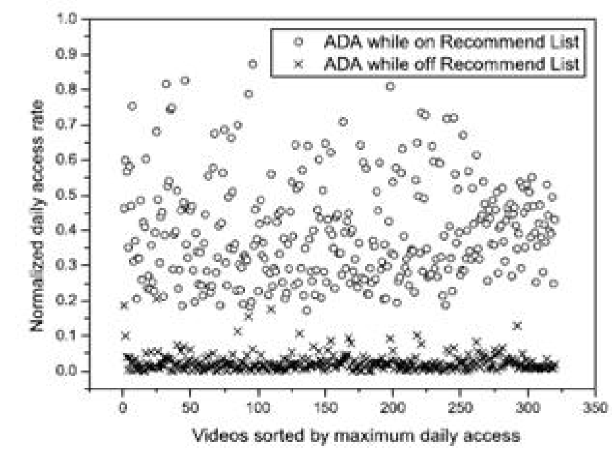


Fig. 4. User interest change over a month [48].

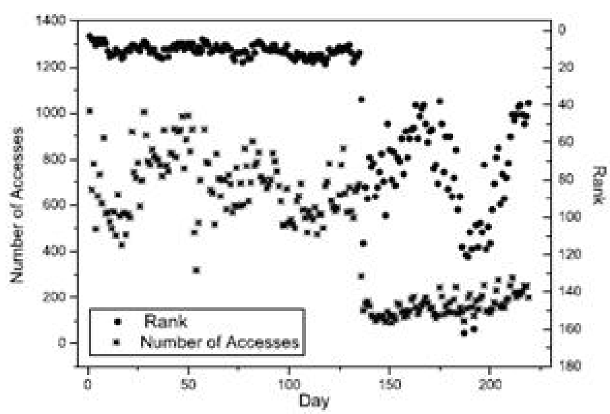


Fig. 5. User interest change over a month [48].

Service provider recommendations may affect the video popularity. Reference [48] compared the normalized average daily access (ADA) rates, where ADA is defined as average daily access rate over maximum daily access rate, for videos which have been on the recommended list. Figure 4 shows that ADA of a video while it is off the recommended list is low (around 5% of maximum rate) and fairly constant, but ADA of the video while it is on the recommended list increases to 20% to 90% of the maximum rate.

Most popular movie listings provided by the service provider may also affect the popularity of a video [48]. Figure 5 shows an example on how a most popular list affects the access frequency of a video; a movie is accessed more frequently while it is on top-15 hot list, but its rank and access frequency drops significantly and cannot recover the popularity once it is out of the top-15 hot list, as shown in the Fig. 5.

3) **Daily Access Pattern:** Daily access pattern, which is related to user behavior, is another important factor in VoD service. As shown in Fig. 6, VoD request pattern for videos during a day has similar pattern, and almost the same for a week: it climbs up to a peak after work (6 pm - 9 pm), and gradually decreases [52].

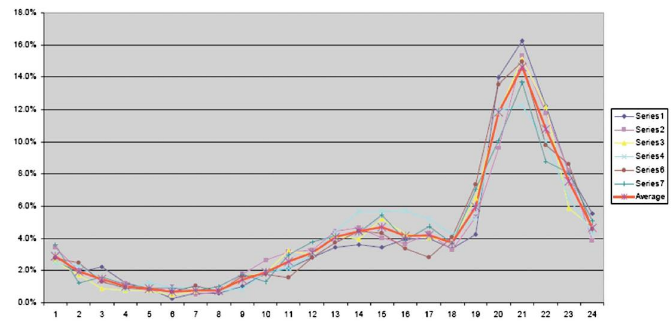


Fig. 6. VoD request change in 24 hours [52].

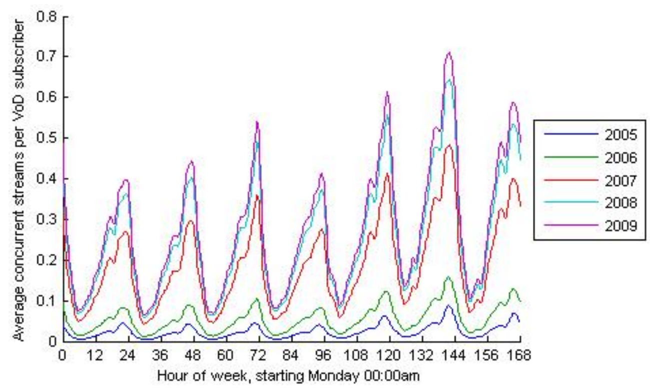


Fig. 7. VoD request change in 7 days [51].

This same pattern is repeated over a week [51] as shown in Fig. 7 and has the highest rate on Saturday [51], [52]. This request pattern may change a little bit based on the working hours in different countries [47], [48], [51], but they all show the same information that users enjoy VoD service during afternoon break and after work. An interesting point to note is that the request rate for videos is quite low during most of the day except between 6pm - midnight, and is highly time-predictable.

From this discussion, it is apparent that newer releases may have frequent access, and schemes that are efficient for deliveries to large number of users should be used. It is also important to consider better network utilization and efficient distribution of videos during the time of the day when the user requests for VoD are highest in volume and frequency.

B. Interactive VCR Behavior

After a user selects a video and starts watching it, the user may find the video uninteresting and the user may employ interactive VCR functions such as Fast Search (FS) before the streaming for the movie is finished. To distinguish a new admission request from an interactive function, the definition of the user interaction in [49] can be used: namely, user session is the time period when a sequence of interactive actions is requested from the same user on the same video, if the time interval of consecutive requests does not exceed a certain threshold. Possible user interactions in a session can be classified as follows [55].

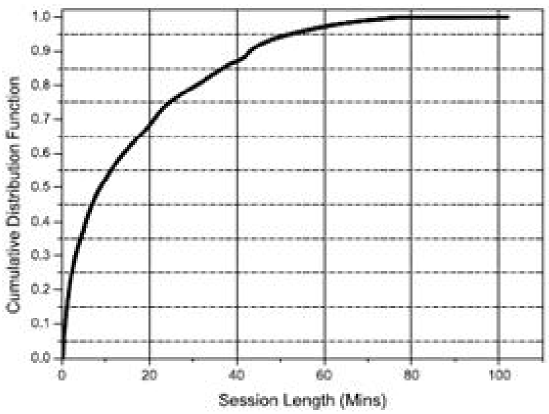


Fig. 8. Session length distribution [48].

- *Play/Resume:* start video playback from the beginning or start playback from other location.
- *Stop/Abort:* stop playback without picture but without disconnecting the connection, or disconnect the connection.
- *Pause:* stop playback with picture.
- *Jump Forward (JF)/Jump Backward (JB):* move to a specific location without picture and sound in forward or backward direction.
- *Fast Search (FS)/Reverse Search (RS):*): move to a specific location with picture and sound in forward or backward direction.
- *Slow Motion (SM):* playback slowly to forward direction with picture and sound.

1) **Session Characteristics:** A user may begin its session from the beginning of a video or in the middle of the video. The user may terminate its session before viewing an entire movie for several reasons. Reference [49] shows that a small but significant fraction of video sessions start at arbitrary positions in the video file, which is quite different from the case of audio where almost 100% of sessions are initiated from the beginning. This phenomenon becomes more pronounced when the video file size increases. The number of interactive actions in a video session also increases with the video file size [44], [49].

Figure 8 shows that session length for videos is quite short: 52.6% of all requests have less than 10 minutes of session length, and 70% of sessions are terminated in the first 20 minutes [48]. This means that a lot of viewers view only the beginning of a video and quickly decide to jump to other videos. Reference [47] also shows that 20%-40% of a movie session is terminated in less than 10 minutes.

One interesting result is that the session length has inverse correlation with video popularity, which means that less-popular videos have longer session length [48]. Reference [48] speculates that users who request a popular movie may have seen it before and lose interest easily. However, a user who requests a less-popular video will have a tendency to really watch that video. The other interesting observation is that session length highly depends on the file size even though some differences are shown in [44], [54].

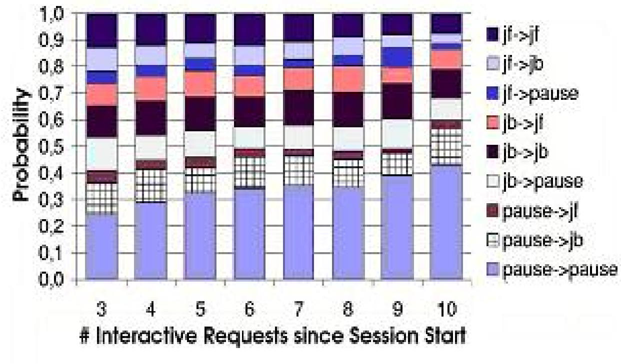


Fig. 9. Probability of consecutive client interactions [49].

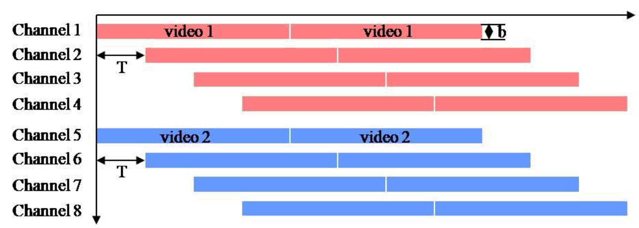


Fig. 10. Staggered broadcasting.

2) **Interactive VCR Operation:** Interactive user behavior pattern is one of the important data for VoD service, but it is not easy to collect the data and build a proper model. Among several interactive operations, pause action is the most common user interaction in VoD service, and jump forward and jump backward actions are equally frequent for long videos [44], [49]. These actions such as pause, jump forward, and jump backward depend on the previous action but not on the number of interactions after the beginning of the session [49]. Figure 9 shows the probability of consecutive interactions, which is collected from high-bit-rate educational contents; for example, from pause action to pause action has the highest probability and is independent of previous interactions. Another interesting property is that any interaction is followed by the same interaction more frequently (e.g., jf $\rightarrow$ jf, pause $\rightarrow$ pause). Reference [49] shows that jump distance, which is the amount of time skipped between two consecutive interactions, increase with file size. Some papers [46], [54], [55], [56] propose interactive behavior models to create more realistic workload.

IV. Bandwidth Saving Multicast Streaming Schemes

When a service provider has an estimation of how a user may behave and interact with a video, it is important for them to facilitate such behaviors and interactions for better customer experience. Bandwidth-saving multicast streaming schemes enable service providers to bring VoD services to their customers in efficient ways. VoD service is inherently a

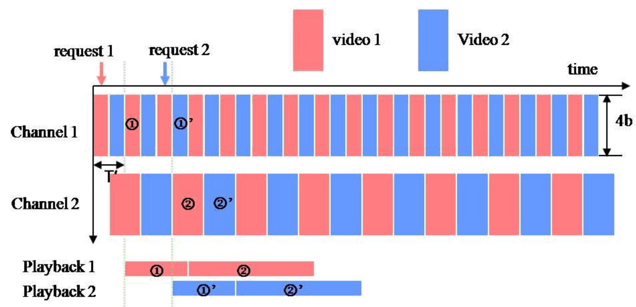


Fig. 11. Pyramid broadcasting.

personalized service and ideally needs a dedicated stream for each customer request. However, to provide a unicast stream for each request is very expensive. Therefore, over the past decade, quite a lot of research has been conducted to provide VoD service economically by using multicast streams. Most well-known and efficient video streaming schemes include broadcasting [4], [5], [6], [7], batching [8], [9], [10], [11], patching [12], [13], [14], and stream merging [14], [15], [16]. Unlike unicast streaming, which uses a dedicated unicast stream for each VoD request, these schemes use a multicast stream and exploit the user's storage equipment. Thus, they can reduce video delivery cost by minimizing the usage of network bandwidth while providing almost the same quality of service to customers.

A. Broadcasting

A simple method to save on bandwidth for VoD service is broadcasting. In broadcasting, the service provider periodically broadcasts a video irrespective of user requests, and customers who want to watch the video can watch it by waiting for the next broadcasting cycle. Therefore, broadcasting scheme can provide near VoD service but it is not a true VoD service.

Staggered broadcasting (SB) [57] is a simple broadcasting protocol to reduce user waiting time in which a video is periodically broadcasted on several channels but with staggered start times. As shown in Fig. 10, video 1 and video 2 are repeatedly broadcast on four channels, each with a staggered time T, so T is the maximum waiting time for a user to receive service. Therefore, the bandwidth requirement of this protocol is proportional to the number of videos rather than the number of users. However, the problem is that the access latency can be improved only linearly by increasing the number of service channels for each video.

An improvement is periodic broadcasting with segmentation where a video file is divided into segments, and each segment is periodically broadcast on dedicated server channels. In this scheme, the size of each segment transmitted on different channels increases geometrically, and each segment is broadcast at a higher rate than playback rate. Figure 11 shows an example of periodic broadcasting which is called pyramid broadcasting [4] where each channel's bandwidth is larger ($4b$ in Fig. 11) than staggered broadcasting (b in Fig. 10). Since broadcast channel bandwidth ($4b$) is larger than playback speed (b), when a segment is playing, other segments are stored in user buffer.

TABLE I
PERIODIC BROADCASTING PROTOCOLS.

	Category 1	Category 2	Category 3
Each segment size for a video	Different (Increasing size)	Equal	Equal
Broadcasting bandwidth for each segment	Same	Different (Decreasing bandwidth)	Equal
Related Protocols	Pyramid Broadcasting (PB), Skyscraper Broadcasting (SkB), Fast Broadcasting (FB)	Harmonic Broadcasting (HB),Cautious Harmonic (CHB)	Pagoda broadcasting

As shown in Fig. 11, if user request 1 comes, the earliest segment that can be approached from the beginning is ① and the next segment is ②. Thus, users download ① and ② sequentially with four times the playback speed ($4b$) and store the video data. Since playback speed (b) is lower than download speed, users can playback without delay from the stored data. In this scheme, users also have to wait for the beginning of the first segment before they can start playing. However, the user waiting time is reduced compared to staggered broadcasting. Maximum user waiting time is usually the length of the first segment (i.e., $T' < T$).

In periodic broadcasting, latency is improved if clients are allowed to preload data. Efficient broadcasting protocols use video segmentation. A video is divided into n segments, and each segment is broadcast on different channels. This kind of periodic broadcasting protocols can be classified into three different categories based on the segment size and broadcast bandwidth for each segment as shown in Table I.

Based on segmentation, there are several schemes that implements periodic broadcasting, such as skyscraper broadcasting (SkB) [5], fast broadcasting (FB) [58], harmonic broadcasting (HB), cautious harmonic broadcasting (CHB), and pagoda broadcasting (PB). In SkB, [5], a video is fragmented into k segments, whose size has [1, 2, 2, 5, 5, 12, 12, 25, 25, ..., W, W, ..., W]. A client has two loaders, and each loader downloads either the odd-numbered group (1, 5, 25, etc.) or the even-numbered group (2, 12, etc.). In FB, a video has segments, whose size follows the geometrical series of [1, 2, 4, , 2k-1, 2k]. In HB and CHB, each segment size for a video is the same, but it is broadcast in a smaller bandwidth than playback rate. The client receives data from all the channels simultaneously. PB [59] is the combination of pyramid broadcasting and harmonic broadcasting, and each segment size for a video and broadcasting bandwidth for each segment has equal value.

B. Batching

Even though broadcasting can reduce the required bandwidth, it is not an efficient method for unpopular videos. Using the previous schemes, broadcast channels for a video would be assigned even though the request rate for that video could be quite low. Therefore, a more efficient method, which can assign channels more flexibly according to the user request

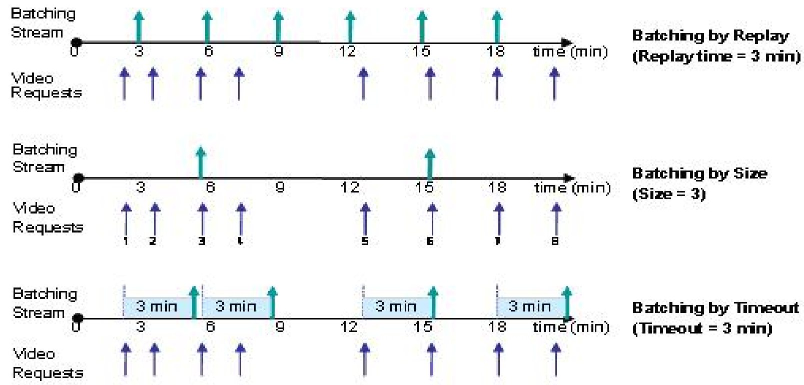


Fig. 12. Three batching policies [8].

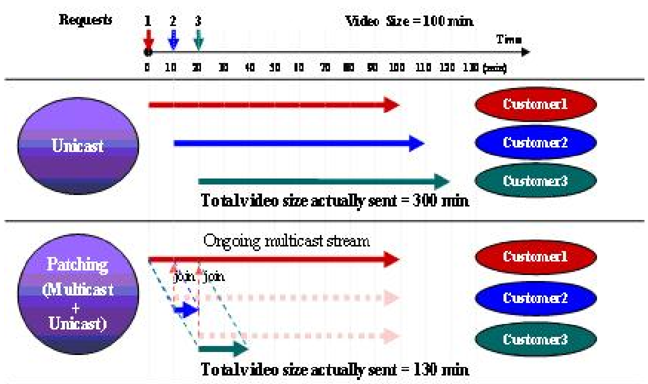


Fig. 13. Comparison of unicast and patching.

rate, is required to improve the shortcomings of broadcasting; one such set of schemes is based on batching.

In batching, multiple client requests for the same video that arrive within a small time period (batching window) are grouped and served using a single multicast transmission channel [8][10]. Therefore, some of the early-arriving requests have to wait until they are served by a grouped stream, so some service delay for an early-arriving request will occur in batching. To reduce the risk of reneging, which means a user cancels its request because of a long waiting time to receive service, it is important to decide on the proper batching window size and strategy. Three kinds of batching policies are known [8]: namely batching-by-replay, batching-by-size, and batching-by-timeout, as shown in Fig. 12.

- *Batching-by-replay:* replays the same video at fixed time intervals. This scheme may waste channel bandwidth when sending a stream without any customers.
- *Batching-by-size:* sends a stream when a certain number of requests have already arrived. This scheme can not guarantee the customer's maximum waiting time since the arrival of requests are not known a priori.
- *Batching-by-timeout:* uses a timer and sends a stream within a specific time after the first request comes. Thus, it can prevent sending a stream without requests as in batching-by-replay, and it can guarantee the waiting time unlike batching-by-size.

Batching is also useful to deal with network congestion, where many requests may be waiting in queues without being served. Thus, several waiting requests for the same video can be served together by batching without additional delay. An important point is to decide which requests should be served first to minimize the overall service delay. Several approaches have been proposed, which apply batching in a server using parallel queues to decide requests to be served first: namely FCFS (First Come First Served), MQL (Maximum Queue Length), and MFQL (Maximum Factored Queue Length); and MFQL shows good overall performance [9].

C. Patching

Whereas broadcasting and batching schemes may save on bandwidth for VoD service, they cannot provide real-time VoD service without significant delay. Patching [12], [13], [14], among various video streaming schemes, is a simple bandwidth-saving approach which can provide real-time service to user requests; in patching, the first request for a video is served by a multicast stream, and subsequent requests for the video are served by initiating a unicast patch stream and by joining the ongoing multicast stream shortly thereafter.

As shown in Fig. 13, subsequent users can receive the later part of the video by joining the multicast stream (dotted red arrow in Fig. 13), but at the same time, they receive the previous part of the video through a separate unicast stream (blue and green solid arrows in Fig. 13). (The multicast stream is cached in the user's local memory for later playback, after the unicast patch stream finishes.) Therefore, only the patch streams contribute to increasing the required server bandwidth for the subsequent requests. Customers can immediately watch the video with the patch stream, and the pre-received multicast stream is stored in user storage for later playback.

To minimize the required bandwidth, a decision is needed for when to provide a multicast stream and when to provide a patch stream for each user request. Two different strategies have been proposed [12]: greedy patching, where a patch stream is generated when a multicast stream exists, or grace patching where a patch stream is generated when a request comes at a specific time window called optimum patching window after a multicast stream is initiated. The specific time window, which is called patching window, is defined as a certain time period after a multicast stream is initiated; if a request arrives in this time window, the request is served by patch streams. Thus, while an earlier video request is being

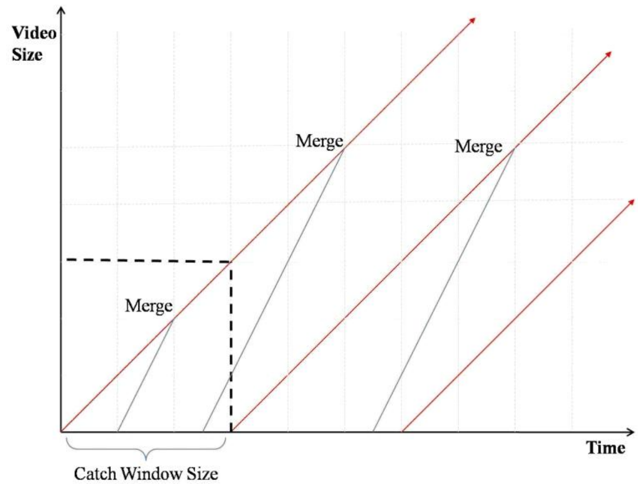


Fig. 14. Adaptive piggybacking [8].

served by a multicast stream, if a subsequent request for the same video arrives during this time window, the request is served by a patch stream for the earlier part of the video and it is also served by joining the ongoing multicast stream for the later part of the video.

D. Stream Merging

In patching, later requests for a video are served by a unicast patch stream and a multicast stream, so subsequent requests can only join one multicast stream. Therefore, if a multicast patch stream is used rather than a unicast patch stream, more bandwidth can be saved by having later requests join several multicast streams. Thus, in merging, all the video streams including patch streams are multicast streams so that any client can join and receive them. Therefore, a new client can receive two or more streams so that a later part of video is received by joining the earlier-initiated multicast streams, and previous part of the video is received by generating another multicast stream [16]. A traditional adaptive piggybacking scheme provides the concept of merging; and, after that, hierarchical stream merging techniques can be used [8], [14], [16].

- *Adaptive piggybacking:* uses the property that two different streams can be merged into one by adjusting the frame rates, since the human eye cannot notice a small change in the video streaming rate [8]. Two streams for the same video request, which are started at slightly different times, can be merged into one after a certain time by increasing the streaming speed of a stream as shown in Fig. 14.
- *Hierarchical stream merging:* attempts to capture the advantages of piggybacking as well as patching. A new client can merge into larger and larger groups, and clients are merged using multicast patch streams [14].

For example, in Fig. 15, requests are initiated by clients ①, ②, ③, and ④ at times 0, 10, 30, and 40, respectively, and each new client request initiates a new stream. Client ④ receives both the stream for ④ and the stream initiated by client ③, and merges with stream ③ at time 50. And this stream again merges with stream ① at time 80.

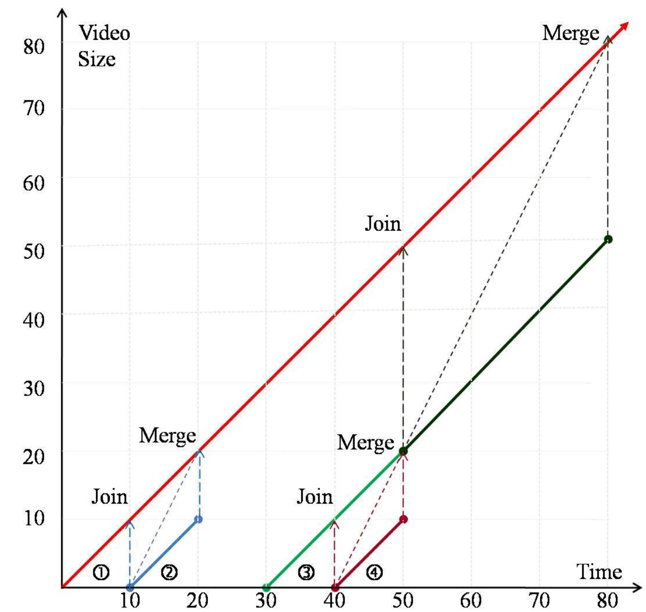


Fig. 15. Example of hierarchical stream merging [14].

E. Recent Hybrid Schemes

Recent research trends in streaming schemes combine the different streaming schemes so that bandwidth usage and user waiting time are minimized, as discussed below.

- *Hybrid transmission scheme [60]:* uses both broadcasting (Fast Broadcast) for the most-popular videos and multicasting (Grace Patching) for the least-popular videos to minimize user waiting time, when a total number of available channels is given for a group of videos with different popularity. Therefore, given a total number of channels, the optimal channel allocation for broadcasting and multicasting is calculated so that user waiting time is minimized.
- *Client-assisted patching [61]:* uses batching, patching, and clients' buffers. Client requests are batched for some time and served by a multicast stream. If a new request comes, the buffer of the client, who already starts receiving the video, is used to patch the missing portion of the subsequent client. The admission controller, which has a centralized database for the entire system, selects a nearby client to supply the patching stream to the subsequent client. Therefore, only one single server stream is needed for the whole group.
- *Reverse fast broadcasting (RFB) [58]:* focuses on client buffer reduction. RFB uses two properties: server transmits the segments in descending order, which is ascending order in FB on each channel, and clients receive segments as late as possible to reduce the buffer requirement of the clients.
- *Fast-start rate control [62]:* proposes a method to transmit at a higher rate at the beginning of the video download since, if there is a network delay, transmitting at $1/T$ frames per second (T is inter-frame time) is not enough to playback without any packet misses. Therefore, a general approach is to buffer some frames for some time

TABLE II
COMPARISON OF VARIOUS STREAMING SCHEMES [63].

Streaming scheme	Immediate playback	Server bandwidth increase to the number of requests	Client bandwidth to the display rate	Client buffer requirement
Batching	No	Logarithmically	Same	No
Patching	Yes	Square root	At least double	High
Piggybacking	Yes	Sub-linear	Same	No
Hierarchical streaming merging	Yes	Logarithmically	At least double	High
Periodic broadcasting	No	No increase	Depends on protocol	High

d_s before playback and to delay the playback time of the first frame.

F. Comparison of Multicast Streaming Schemes

Major characteristics of the above schemes, which include batching, piggybacking, patching, hierarchical stream merging, and periodic broadcasting to reduce bandwidth, are compared in Table 2 [63]: Batching and broadcasting are efficient streaming schemes in the aspect of server bandwidth. But these schemes cannot provide true VoD service but only near VoD service, since customers experience delay for the request to be served.

To provide true VoD service, patching and hierarchical stream merging are proper solutions to be considered. Among the two schemes, hierarchical stream merging is more efficient in bandwidth usage, but it has the drawback of higher client bandwidth, and higher buffer requirement which make the system and user equipments quite complex to achieve good performance.

V. INTERACTIVE VCR FUNCTION SUPPORT IN MULTICAST STREAMING SCHEMES

The streaming schemes described in Section III can reduce the required bandwidth for VoD service by using multicast streams and exploiting user buffers. However, these methods – although efficient from a bandwidth-usage perspective – create complexity with respect to user VCR interaction and decrease the efficiency significantly: e.g., if a user who operates interactive actions breaks away from a multicast stream, which several other requests are sharing, and has to be served by a separate stream for the VCR action. Reference [64] shows that even new multicast streaming schemes to support interactive VCR functions are not efficient on bandwidth usage; for example, the interactive playback latency increases to 100s of seconds from 10s of seconds when mean interactive playback request is less than 1 request/client [64]. This phenomenon occurs because contingency channels, dedicated unicast channels, which are required for continuous VCR function, significantly increase the system bandwidth.

A. Classification of VCR Interactivity

The methods for VCR support can be classified into two categories [66]: discontinuous VCR support and continuous VCR support. In discontinuous VCR support, only limited VCR actions can be provided to users, e.g., users can operate VCR actions continuously only for predetermined time intervals. However, in continuous VCR support, users can freely operate any VCR functions for any duration. VoD systems can be categorized as follows based on the ability to control videos by the user and the level of interactivity [67][68]: No-VoD (Broadcast), Pay-Per-View (PPV), Quasi Video-on-Demand (QVoD), Near Video-on-Demand (NVoD), and True Video-on-Demand (TVoD). Among them, NVoD and TVoD, which are of interest to us, can be achieved by using previous multicast streaming schemes; NVoD supports discontinuous VCR actions and TVoD support continuous VCR functions.

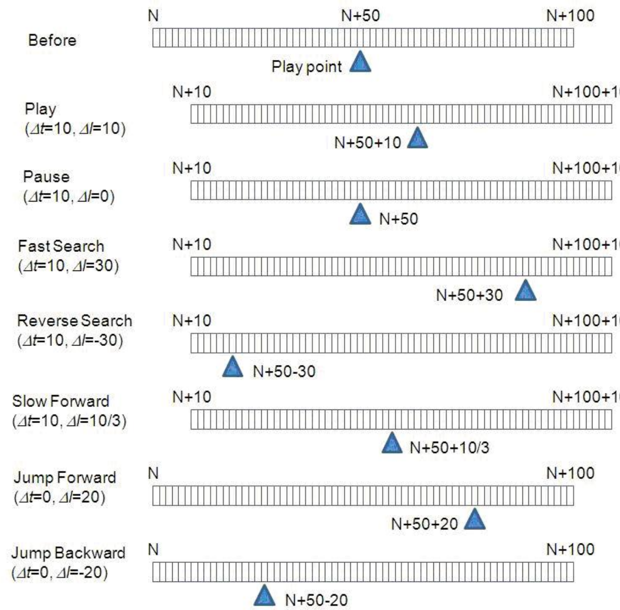


Fig. 16. Change of play position in a user buffer [65].

B. Principles for VCR Support in Multicast Streaming Schemes

To support interactive VCR actions such as Fast Search (FS) and Reverse Search (RS) or even Pause/Resume in multicast streaming schemes, some amount of pre-downloaded video data is needed. Pause/Resume, for example, can be supported by storing video data from the point that Pause action occurs to continue playback when Resume action is generated. In other words, when a user resumes playback after Pause, the ongoing multicast stream has already passed the restarting point and the user cannot begin playback by joining the original multicast stream again. So, the user needs to buffer video data from pause point, which makes it possible to continue future playback.

However, if the user buffer space is small and there is no proper multicast stream to join after consuming all the data in the buffer, continuous playback may not be possible after Resume. It is also not possible to support continuous FS or RS actions with only a limited amount of buffer space. Therefore, one research direction is how to efficiently support VCR functions with limited amount of user storage. However, only efficient buffer management may not be able to support continuous VCR actions. So another method, which can provide continuous interactive functions by using another separate channel for VCR support, is proposed. In this method, if an interactive action is requested while a user is playing a video from a user buffer, the action is served by a dedicated channel allocated only for the action. This mechanism works as follows: first, the play point of the VCR action is checked to determine if the interactive action can be served from the stored data in user buffer; second, if the play point is beyond the stored data in the buffer, the request is served by allocating another unicast channel to continue the VCR function.

Thus, VCR support mechanisms fall into two categories: 1) limited or discontinuous VCR support by using efficient buffering strategy, and 2) full or continuous VCR support by allocating a contingency channel, for the VCR action.

1) **Caching Video Data in User Buffer for Discontinuous VCR Support:** Suppose we have a limited capacity of user buffer and some of the data are continuously cached from multicast streams and discarded some time after playback. If a user requests a VCR action while he/she is playing the video, the play point in the buffer is changed as shown in Fig. 16 (N is the frame number of a video, Δt is the time taken for the VCR action, and Δl is the relative position from the current play point after the VCR action for the time Δt) [65]. Consider the case where buffered data is from N to N+100 and current play point is N+50 as shown in Fig. 16. If a user plays a video for Δt=10, the buffered data is changed from N+10 to N+100+10, and the play point is also changed to N+50+10 for the time t. If the user generates an FS action for Δt=10, the buffered data is the same as the case of play, which is from N+10 to N+100+10. However, the play point of the buffer goes from N+50 to N+50+30, since Δl = 30.

   As expected from this example, if a user generates a FS action for a longer time, which is beyond the buffered data, then it is not possible to support FS action continuously. Therefore, to support VCR actions more efficiently by preventing VCR actions from going beyond the buffered data as much as possible, an efficient play-point management scheme, such as Active Buffer Management (ABM) [65] is needed. How an ABM manages buffers efficiently to accommodate VCR functionalities is discussed in Section V-C.

2) **Allocating Contingency Channel for Continuous VCR Support:** More positive approaches, which provide true VoD service by supporting continuous VCR actions, have also been researched. The idea of these approaches is to allocate a new dedicated channel for VCR support.

   The basic mechanism of these schemes can be summarized in two steps: interaction (split from multicast stream) process and merge process as in [56][69].

   - *Interaction process:* if the data in user buffer can support a VCR action, no additional channel is required. If the user interaction exceeds the capacity of user buffer data, a dedicated channel is assigned to support the VCR action.

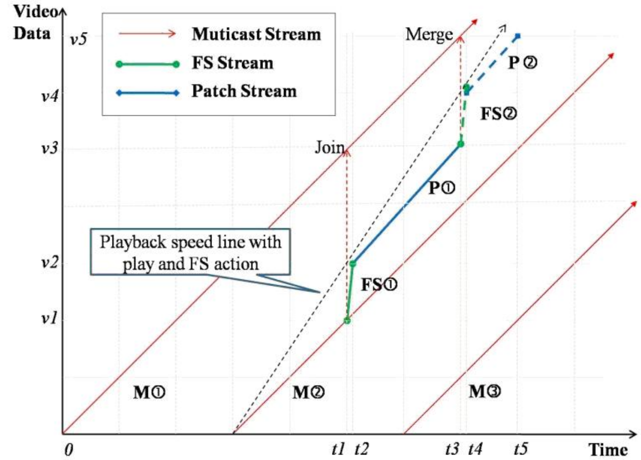


Fig. 17. Example for allocating contingency channels for FS action.

   - *Merging process:* after the VCR action, if the restarting point is in the user buffer, no additional stream is required. But if the restarting point is not in the user buffer, another stream to join the ongoing multicast stream is required.

Let us study an example of FS support in a patching scheme using the above two steps in Fig 17. Suppose a user is playing a video by joining a multicast stream M②; and at time $t1$, FS action is requested. If there is no buffered data in user storage, a dedicated stream to serve the FS action (FS① is needed as shown in Fig. 17). If the FS action stops and resumes playback at $t2$, this dedicated stream for FS action continues until time $t2$. After FS action, playback is resumed. Since this request cannot join the original multicast stream (M② in Fig. 17) after the FS action, it should find a closest multicast stream to join (M① in Fig. 17). To join the multicast stream M① after FS action, another patch stream P① is needed and it merges with the multicast stream M① as shown in Fig. 17. If another FS action is generated after merging to multicast stream M①, since video data from $v3$ to $v5$ is already stored in user's storage, FS action FS② can be supported and playback P② directly from user storage as shown by the dotted lines FS② and P② in Fig. 17. After the playback speed line crosses with multicast stream M①, a request for FS action repeats the same process until it finishes streaming.

Previous research efforts, while providing continuous VCR support using contingency channels, mainly focused on optimizing the required contingency channels and tried to minimize the required bandwidth by reducing the additional unicast channels for VCR support [64], [70], [71], [72].

C. VCR Support in Different Multicast Streaming Schemes

Multicast streaming schemes can save on network bandwidth for serving user requests, but their major challenge is the difficulty of VCR function support. To address this problem, mechanisms have been developed to support VCR functions in VoD systems. Some VCR support schemes, which

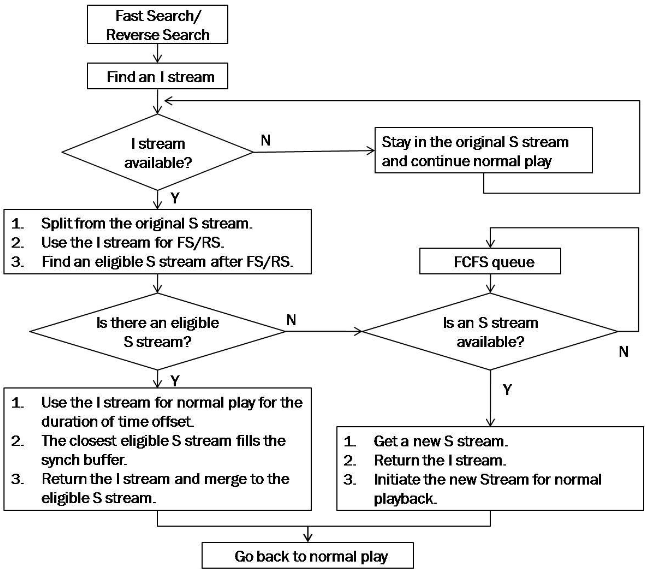


Fig. 18. Flowchart for the fast-forward and rewind operation [56].

consider batching as basic multicast streaming technology, are proposed in [56], [66], [73]. Reference [66] supports continuous and discontinuous VCR actions by focusing on pause, fast-forward, and rewind actions using the user's memory (set-top box). In continuous case, user interaction function is fully supported for the duration of requested time. But in discontinuous case, only predetermined duration of interaction time is supported. For example, users can only move forward the play point to 3, 6, 9, ... min (as an example) in fast-forward in discontinuous interaction. This study shows that only continuous pause action can be supported, and continuous fast-forward and rewind functions cannot be supported with only user buffer. And it proposes the necessity of emergency channels to provide continuous interactive functions.

Reference [56] proposes a split-and-merge (SAM) protocol, which can provide continuous VCR actions using interactive channels and synch buffer shared by several users. SAM describes the detailed mechanism of most of the interactive functions including Pause, Fast Forward/Reverse, and Jump Forward/Backward. Its basic idea is to use a separate interactive stream to provide a VCR action. So this approach divides the system streams into two types: Service stream (S stream) for normal playback and Interaction stream (I stream) for interactive function.

Figure 18 shows the process of supporting FF and Rewind action. If a user requests FF action, SAM must first find an I stream and split the user from the original S stream to serve the FF action. After FF action, to merge to S stream, the user uses I stream for normal playback until it merges to S stream again. Since one I stream for one user is allocated, SAM can guarantee the user interaction support after VoD service begins. The work in [73] improves the SAM protocol and makes the system more scalable by changing the shared synch buffer to separate user buffer to pre-fetch frames to continue playback by merging back to S channel after interaction.

In the above schemes, batching is considered as the multicast streaming technology. Other schemes using broadcasting are also proposed in [65], [57], [74], [75], [76].

The works in [65], [76] propose ABM in a broadcast VoD system, which pre-downloads video segments from several broadcast channels by considering the play point of the user buffer. Since VCR actions such as FS may be performed consecutively, the play point in a user buffer could go to the end of the buffer, and this could make it difficult to provide continuous interactions. To address this problem, ABM controls the stored data by receiving data from "past", "present", and "future" streams after VCR action occurs so that the play point stays in the middle of the buffer. By repeating this process, ABM can lower the probability of the VCR action by moving the play point out of the buffer space. However, if the buffer space is limited, ABM cannot guarantee the continuous interactive functions; Ref. [57] proposes a reception scheduling method for candidate frames in staggered broadcasting by analyzing the conditions for consistent VCR actions.

In [75], the authors propose a method which optimizes the ABM scheme with contingency channel allocation to support interactive functions. In a true VoD system, only buffer management is insufficient, and contingency channel should also be used to serve continuous interactivity. However, since the increase of contingency channels may degrade the system performance, efficient management of contingency channels is necessary. This study develops a greedy channel management scheme (GCM) to minimize the bandwidth for contingency channels. It first considers pre-fetched data in a buffer in ABM and only transmits the missing data for users to merge to broadcasting channels. This scheme minimizes the admission delay by using the periodic broadcasting scheme as described in Section II instead of the staggered broadcasting scheme.

Reference [74] proposes a different approach which uses video display property for interactive functions since FF action, for example, does not require all the frames for the normal playback but requires only sparse frames or interactive version of frames. Therefore, this work proposes to broadcast interactive version of videos to support VCR action together with normal broadcasting channels.

The above schemes basically use batching and broadcasting as multicast streaming technologies, and these multicast streaming schemes inherently support only NVoD service which has some admission delay. Therefore, even though continuous interaction functions are supported, they eventually support only NVoD service. Thus, other research efforts to use different multicasting scheme such as patching, which does not incur admission delay, have also been conducted.

More recent work, called Best-Effort Patching (BEP) [69], provides true VoD service in the aspect of both request admission delay and VCR interaction support. In BEP, if a user executes VCR actions and the interaction exceeds the capacity of user buffer, a separate patch stream is provided to serve the interaction and a merge operation is also used. The data from the patch channel when interaction is supported by patch stream is immediately played back, and the data from multicast stream is stored in user buffer. For merge process, patch stream is required to join the closest ongoing multicast stream.

The work in [64] shows the performance degradation of modern multicast streams when interactive VCR functions

TABLE III
MUTICAST SCHEMES AND THEIR VCR SUPPORT MECHANISM.

Multicast scheme	Reference (Protocol)	Supported Interactivity	Support Mechanism	Features
Batching	[29](SAM)	Continuous	Interactive channel Shared synch buffer	Describes most of the VCR action support mechanism by S stream and I stream
	[36]	Continuous Discontinuous	Emergency channel User buffer	Pause, Fast Forward, Rewind actions
	[44]	Continuous	Interactive channel Separate user buffer	Improves SAM protocol using separate user buffer and pre-fetch frames
Broadcasting	[39][48] (ABM)	Discontinuous	Active buffer management	Receives video data from past, present, and future streams after VCR action occurs. Manages the play point to stay in the middle of buffer
	[45]	Discontinuous	Reception scheduling	Schedules receiving candidate frames for consistent VCR actions
	[46]	Discontinuous	Interactive version of videos	Broadcasts interactive version of videos to support VCR action with normal broadcasting channels
	[47]	Continuous	Contingency channel	Applies ABM and transmit missing data for users to merge into broadcasting channel
Patching	[40](BEP)	Continuous	Patch stream	Patch stream is used to serve interaction support and merge back to ongoing a multicast stream
	[35] (SFSS)	Continuous	Patch stream	Full-stream restart threshold is used to minimize performance degradation from increasing merge requests

are considered, and it proposes a scheme to control a full-stream restart threshold. This approach divides requests into (1) admission requests which are generated by newly-admitted requests; and (2) merging requests which are generated by interactive functions. These requests are stored in two separate queues to be scheduled, and they are optimally scheduled by full-stream restart threshold. In other words, an admitting request is served every full-stream threshold; otherwise, merging requests are served. Since delays in merging requests are more critical than those of admission requests, priority can be given to merging requests when bandwidth is available. However, this method makes inefficient resource usage when merging requests increase but full streams decreases, which consequently prevents merging requests from merging to full streams. Therefore, in [64], the authors propose algorithms, called Static Full-Stream Scheduling (SFSS) and Just-In-Time Simulation (JTS) Schemes, which calculate and control the full-stream restart threshold to improve the performance. Table III shows the comparison of several multicast streaming schemes and VCR function support mechanisms.

VI. CONCLUSION

Since VoD service is a very promising service, the service provider needs to consider cost-effective methods for service deployment. In this paper, we reviewed important factors in VoD service such as user behavior and bandwidth-saving multicast schemes, which are important for properly designing the service architecture and related mechanisms. We first reviewed the user behavior such as video access pattern, how it changes over time, and user VCR interaction, and how these information can be used to build a user-centric VoD service network architecture. We also reviewed the bandwidth-saving multicast streaming schemes to help optimize the usage of given resources, which has been one of the popular research topics. We also reviewed the principles of VCR functions in multicast streaming schemes and different proposals for different multicast streaming schemes. In this survey, we presented how video files are accessed by the clients for VoD services and how they can be delivered to the viewers with maximum efficiency based on the scenarios. Through this survey, a well-informed concept can be developed regarding the VoD services and their deployment choices.

REFERENCES

[1] http://broadband.motorola.com/dvr/dct6412.asp.

[2] Research and markets, "Market monitor: cross-platform video on-demand," Feb. 2009.

[3] Ovum, "Big picture: VoD world revenues 2007-2011," May 2007.

[4] S. Viswanathan and T. Imielinski, "Metropolitan area video-on-demand service using pyramid broadcasting," *Multimedia Systems*, vol. 4, pp. 179–208, Aug. 1996.

[5] K. A. Hua and S. Sheu, "Skyscraper broadcasting: a new broadcasting scheme for metropolitan video-on-demand systems," in *Proc. ACM SIGCOMM*, Cannes, France, pp. 89–99, Sep. 1997.

[6] A. Hu, "Video-on-demand broadcasting protocols: a comprehensive study," in *Proc. IEEE INFOCOM*, pp. 508–517, Sep. 2001.

[7] R. M. Zafalao and N. L. S. da Fonseca, "Periodic broadcasting protocols for clients with bandwidth limitation," in *Proc. IEEE Globecom*, Nov. 2006.

[8] H. J. Kim and Y. Zhu, "Channel allocation problem in VoD system using both batching and adaptive piggybacking," *IEEE Trans. Consum. Electron.*, vol. 44, no. 3, pp. 969–976, Aug. 1998.

[9] C. C. Aggarwal and J. L. Wolf, "On optimal batching policies for video-on-demand storage servers," in *Proc. IEEE Int. Conf. on Multimedia Computing and Systems*, Hiroshima, Japan, June 1996.

[10] V. Tokekar, A. K. Ramani, and S. Tokekar, "Analysis of batching policy in view of user reneging in VoD system," in *Proc. INDICON*, Dec. 2005.

[11] N. L. S. da Fonseca and R. D. A. Facanha, "The look-ahead-maximize-batch batching policy," *IEEE Trans. Multimedia*, vol. 4, no. 1, pp. 114–120, Mar. 2002.

[12] K. A. Hua, Y. Cai, and S. Sheu, "Patching: a multicast technique for true video-on-demand services," in *Proc. ACM Int. Conf. Multimedia*, pp. 191–200, Sep. 1998.

[13] Y. Cai, K. A. Hua, and K. Vu, "Optimizing patching performance," in *Proc. IS&T/SPIE Conf. on Multimedia Computing and Networking*, Jan. 1999.

[14] D. Eager, M. Vernon, and J. Zahorjan, "Minimizing bandwidth requirements for on-demand data delivery," *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 5, Sep./Oct. 2001.

[15] R. T. Shi, L. Shao, Y. Z. Pei, and D. Xie, "A novel stream merging algorithm for VoD servers," in *Proc. ICICS-PCM*, Dec. 2003.

[16] D. Eager, M. Vernon, and J. Zahorjan, "Optimal and efficient merging schedules for video-on-demand servers," in *Proc. Int. Multimedia Conf.*, 1999.

[17] H. Yu, H. Yang, Y. Wang, P. Fan, and C. Chien, "Broadcasting scheme with low client buffers and bandwidths for video-on-demand applications," *Multimedia Tools and Applications*, vol. 42, no. 3, pp. 295–316, May 2009.

[18] O. Hillestad, A. Perkis, V Genc, S. Murphy, and J. Murphy, "Delivery of on-demand video services in rural areas via IEEE 802.16 broadband wireless access networks," in *Proc. IEEE/ACM International Workshop on Wireless Multimedia Networking and Performance Modeling*, Terromolinos, Spain, pp. 43–52, Oct. 2006.

[19] J. Ding, C. Lin, and S. Lan, "A unified approach to heterogeneous video-on-demand broadcasting," *IEEE Trans. Broadcast.*, vol. 54, no. 1, pp. 14–23, Mar. 2008.

[20] T. Do, K. Hua, N. Jiang, and F. Liu, "PatchPeer: A scalable video-on-demand streaming system in hybrid wireless mobile peer-to-peer networks," *Peer-to-Peer Networking and Applications*, vol. 2, no. 3, pp. 182–201, Sep. 2009.

[21] F. Xie and K. Hua, "A caching-based video-on-demand service in wireless relay networks," in *Proc. IEEE International Conference on Wireless Communications and Signal Processing*, Nanjing, China, pp. 1–5, Nov. 2009.

[22] H. Dai and E. Chan, "Quick patching: an overlay multicast scheme for supporting video on demand in wireless networks," *Multimedia Tools and Applications*, vol. 36, no. 3, pp. 221–242, Feb. 2008.

[23] C. Cho, K. Kim, H. Youn, and S. Hwang, "An effective batching scheme utilizing the segment interval time for wireless IPTV," in *Proc. IEEE Conference on Consumer Communications and Networking Conference*, Las Vegas, NV, pp. 442–446, Jan. 2009.

[24] S. Misra, M. Reisslein, and G. Xue, "A survey of multimedia streaming in wireless sensor networks," *IEEE Commun. Surveys & Tutorials*, vol. 10, no. 4, pp. 18–39, 4th Qtr. 2008.

[25] B. Cheng, X. Liu, Z. Zhang, H. Jin, L. Stein, and X. Liao, "Evaluation and optimization of a peer-to-peer video-on-demand system," *J. Systems Architecture*, vol. 54, no. 7, pp. 651–663, July 2008.

[26] W. Yiu, X. Jin, and S. Chan, "VMesh: Distributed segment storage for peer-to-peer interactive video streaming," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 9, pp. 1717–1731, Dec. 2007.

[27] Y. Tian, D. Wu, and K. Ng, "A novel caching mechanism for peer-to-peer based media-on-demand streaming," *J. Systems Architecture*, vol. 54, no. 1-2, pp. 55–69, Feb. 2008.

[28] K. Huguenin, A. Kermarrec, V. Rai, and M. Van Steen, "Designing a tit-for-tat based peer-to-peer video-on-demand system," in *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video*, Amsterdam, Netherlands, pp. 93–98, June 2010.

[29] W. Yiu, X. Jin, and S. Chan, "Efficient search and scheduling in P2P-based media-on-demand streaming service," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 1, pp. 119–130, Jan. 2007.

[30] B. Cheng, X. Liu, Z. Zhang, and H. Jin, "A measurement study of a peer-to-peer video-on-demand system," in *Proc. International Workshop on Peer-to-Peer Systems*, Bellevue, WA, Feb. 2007.

[31] B. Cheng, H. Jin, and X. Liao, "RINDY: A ring based overlay network for peer-to-peer on-demand streaming," *Lecture Notes in Computer Science*, vol. 4159, pp. 1048–1058, 2006.

[32] Y. Guo, K. Suh, J. Kurose, and D. Towsley, "P2Cast: peer-to-peer patching for video on demand service," *Multimedia Tools and Applications*, vol. 33, no. 2, pp. 109–129, May 2007.

[33] B. Cheng, H. Jin, and X. Liao, "Supporting VCR functions in P2P VoD services using ring-assisted overlays," in *Proc. IEEE ICC*, Glasgow, UK, pp. 1698–1703, June 2007.

[34] N. Vratonjic, P. Gupta, N. Knezevic, D. Kostic, and A. Rowstron, "Enabling DVD-like features in P2P video-on-demand systems," in *Proc. Peer-to-Peer Streaming and IP-TV Workshop*, Kyoto, Japan, pp. 329–334, Aug. 2007.

[35] T. Do, K. Hua, and M. Tantaoui, "Robust video-on-demand streaming in peer-to-peer environments," *Computer Communications*, vol. 31, no. 3, pp. 506–519, Feb. 2008.

[36] K. Sato, M. Katsumoto, and T. Miki, "P2MVOD: peer-to-peer mobile video on-demand," in *Proc. International Conference Advanced Communication Technology*, Phoenix Park, AZ, pp. 1866–1870, Feb. 2006.

[37] M. Chesire, A. Wolman, G. Voelker, and H. Levy, "Measurement and analysis of a streaming-media workload," in *Proc. Conference on USENIX Symposium on Internet Technologies and Systems*, San Francisco, CA, p. 1, Mar. 2001.

[38] K. Sripanidkulchai, B. Maggs, and H. Zhang, "An analysis of live streaming workloads on the Internet," in *Proc. ACM SIGCOMM Conference on Internet Measurement*, Taormina, Italy, pp. 41–54, Oct. 2004.

[39] L. Cherkasova and M. Gupta, "Characterizing locality, evolution, and life span of accesses in enterprise media server workloads," in *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video*, Miami, FL, pp. 33–42, May 2002.

[40] C. Huang, J. Li, and K. Ross, "Can internet video-on-demand be profitable?" in *Proc. ACM SIGCOMM*, Kyoto, Japan, pp. 133–144, Aug. 2007.

[41] Y. Challal, H. Bettahar, and A. Bouabdallah, "A taxonomy of multicast data origin authentication: Issues and solutions," *IEEE Commun. Surveys & Tutorials*, vol. 6, no. 3, pp. 34–57, 3rd Qtr. 2004.

[42] A. Matrawy and I. Lambadaris, "A survey of congestion control schemes for multicast video applications," *IEEE Commun. Surveys & Tutorials*, vol. 5, no. 2, pp. 22–31, 4th Qtr. 2003.

[43] P. Seeling, M. Reisslein, and B. Kulapala, "Network performance evaluation using frame size and quality traces of single-layer and two-layer video: A tutorial," *IEEE Commun. Surveys & Tutorials*, vol. 6, no. 3, pp. 58–78, 3rd Qtr. 2004.

[44] J. M. Almeida, J. Krueger, D. L. Eager, and M. K. Vernon, "Analysis of educational media server workloads," in *Proc. NOSSDAV*, June 2001.

[45] J. Almeida, J. Krueger, and M. Vernon, "Characterization of user access to streaming media files," in *Proc. ACM SIGMETRICS*, 2001.

[46] P. Branch, G. Egan, and B. Tonkin, "Modeling interactive behavior of a video based multimedia system," in *Proc. IEEE ICC*, June 1999.

[47] M. Vilas, X. G. Paneda, R. Garcia, D. Melendi, and V. G. Garcia, "User behavior analysis of a video-on-demand service with a wide variety of subjects and lengths," in *Proc. EUROMICRO*, Aug./Sept. 2005.

[48] H. Yu, D. Zheng, B. Y. Zhao, and W Zheng, "Understanding user behavior in large-scale video-on-demand systems," in *Proc. EuroSys*, Oct. 2006.

[49] C. Cristiano, I. Cunha, A. Borges, C. Ramos, M. Rocha, J. Almeida, and B. Riberio-Neto, "Analyzing client interactivity in streaming media," in *Proc. WWW*, 2004.

[50] F. Thouin and M. Coates, "Video-on-demand networks: Design approaches and future challenges," *IEEE Network*, vol. 21, no. 2, Mar./Apr. 2007.

[51] K. Couch, "Raising the bar for triple play with VoD," *Converge! Network Digest*, Jan. 2005.

[52] J. Shreeram, "VoD everywhere! Considerations in transport methods for scalable VOD/SVOD deployment," *Scientific-Atlanta, Inc.*, 2002.

[53] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan, "Measurement, modeling, and analysis of a peer-to-peer file-sharing workload," in *Proc. ACM Symp. Operating Systems Principles*, Bolton Landing, NY, Oct. 2003.

[54] W. Tang, Y. Fu, L. Cherkasova, and A. Vahdat, "Modeling and generating realistic streaming media server workloads," *Computer Networks*, vol. 51, no. 1, Jan. 2007.

[55] V. O. K. Li, W. Liao, X. Qiu, and E. W. M. Wong, "Performance model of interactive video-on-demand systems," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 6, Aug. 1996.

[56] W. Liao and V. O. K. Li, "The split and merge protocol for interactive video-on-demand," *IEEE Mulimedia*, vol. 4, no. 4, pp. 51–62, Oct. 1997.

[57] J. B. Kwon and H. Y. Yeom, "Providing VCR functionality in staggered video broadcasting," *IEEE Trans. Consum. Electron.*, vol. 48, no. 1, pp. 41–48, Feb. 2002.

[58] H. Yu, H. Yang, and L. Tseng, "Reverse fast broadcasting (RFB) for video-on-demand applications," *IEEE Trans. Broadcast.*, vol. 53, pp. 103–111, Mar. 2007.

[59] Z. Zhang, H. Kim, and K. Chon, "A hybrid pagoda broadcasting protocol: fixed-delay pagoda broadcasting protocol with partial preloading," in *Proc. International Conference on Multimedia and Expo*, Baltimore, MD, pp. 801–804, July 2003.

[60] S. A. Azad and M. Murshed, "An efficient transmission scheme for minimizing user waiting time in video-on-demand systems," *IEEE Commun. Lett.*, vol. 11, no. 3, pp. 285–287, Mar. 2007.

[61] S. M. Farhad and M. M. Akbar, "Multicast video-on-demand service in an enterprise network with client-assisted patching," in *Proc. PacRim Conference*, Aug. 2007.

[62] Y. Fallah, S. Zahir, and H. Alnuweiri, "A fast-start rate control mechanism for video streaming applications," in *Proc. ICCE*, Jan. 2005.

[63] N. L. S. de Fonseca and H. K. S. Rubinsztejn, "Dimensioning the capacity of true video-on-demand servers," *IEEE Trans. Multimedia*, vol. 7, pp. 932–941, Oct. 2005.

[64] Y. W. Wong, J. Y. B. Lee, V. O. K. Li, and G. S. H. Chan, "Supporting interactive video-on-demand with adaptive multicast streaming," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 2, Feb. 2007.

[65] Z. Fei, M. H. Ammar, I. Kamel, and S. Mukherjee, "An active buffer management technique for providing interactive functions in broadcast video-on-demand systems," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 942–950, Oct. 2005.

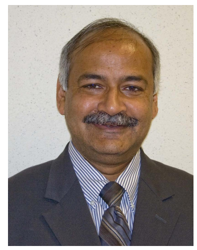
[66] K. C. Almeroth and M. Ammar, "The use of multicast delivery to provide a scalable and interactive video-on-demand service," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 6, pp. 1110–1122, Aug. 1996.

[67] H. Ma and K. G. Shin, "Multicast video-on-demand services," in *Proc. ACM SIGCOMM*, Jan. 2002.

[68] T. D. C. Little and D. Venkatesh, "Prospects for interactive video-on-demand systems," *IEEE Multimedia*, vol. 1, no. 3, no. 3, pp. 14–24, 1994.
[69] H. Ma, K. Shin, and W. Wu, "Best-effort patching for multicast true VoD service," *Multimedia Tools and Applications*, vol. 26, no. 1, May 2005.
[70] M. Y. Y. Leung, J. C. S. Lui, and L. Golubchik, "Use of analytical performance models for system sizing and resource allocation in interactive video-on-demand systems employing data sharing techniques," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 3, May/June 2002.
[71] S.-W. Lee, K.-W. Kim, J.-M. Choi, and K.-D. Chung, "An alternative multicast transmission scheme for VCR operations in a large VoD system," *Lecture Notes in Computer Science*, vol. 2344, 2002.
[72] A. Dan, P. Shahaduddin, D. Sitaram, and D. Towsley, "Channel allocation under batching and VCR control in movie-on-demand servers," *J. Parallel and Distributed Computing*, vol. 30, 1995.
[73] E. L. Abram-Profeta and K. G. Shin, "Providing unrestricted VCR functions in multicast video-on-demand servers," in *Proc. IEEE Int. Conf. on Multimedia Computing and Systems*, June/July 1998.
[74] M. A. Tantaoui, K. A. Hua, and S. Sheu, "A broadcast technique for providing better VCR-like interactions in a periodic broadcast environment," in *Proc. IEEE Globecom*, Nov. 2002.
[75] W.-F. Poon, K.-T. Lo, and J. Feng, "Provision of continuous VCR functions in interactive broadcast VoD systems," *IEEE Trans. Broadcast.*, vol. 51, no. 4, Dec. 2005.
[76] Z. Fei, K. I. Kamel, S. Mukherjee, and M. H. Ammar, "Providing interactive functions for staggered multicast near video-on-demand systems," in *Proc. IEEE International Conference on Multimedia Computing and Systems*, June 1999.

Joonho Choi was born in South Korea in 1967. He received the B.S degree in Electronic Engineering from Yonsei University, South Korea in 1991 and M.S degree and Ph.D. degree in Computer Science from UC Davis, CA in 2008 and 2009 respectively. In 1996, joined Korea Patent Office as a patent examiner and from 1999 to 2005, he has been with the Ministry of Information and Communication. He is currently a Director of International Cooperation Bureau for Korea Communications Commission. His research interests include high speed communication networks, multimedia systems, and user behavior modeling.

Abu (Sayeem) Reaz received the BS Degree in Computer Science from North South University, Dhaka, Bangladesh, in 2003 and the MS Degree in Computer Science from the University of Oklahoma, Norman, USA in 2006. He is currently a PhD candidate in the Department of Computer Science at University of California, Davis. His current research interests include design, development, and routing over Wireless Mesh Network and Wireless-Optical Broadband Access Network.



Biswanath Mukherjee holds the Child Family Endowed Chair Professorship at University of California, Davis, where he has been since 1987, and served as Chairman of the Department of Computer Science during 1997 to 2000. He received the B.Tech. (Hons) degree from Indian Institute of Technology, Kharagpur, in 1980, and the PhD degree from University of Washington, Seattle, in 1987.

He served as Technical Program Co-Chair of the Optical Fiber Communications (OFC) Conference 2009, and is serving as General Co-Chair of OFC 2011. He served as the Technical Program Chair of the IEEE INFOCOM '96 conference. He is Editor of Springer's Optical Networks Book Series. He serves or has served on the editorial boards of eight journals, most notably IEEE/ACM Transactions on Networking and IEEE Network. He is Steering Committee Chair of the IEEE Advanced Networks and Telecom Systems (ANTS) Conference, and served as General Co-Chair of ANTS in 2007 and 2008.

He is co-winner of the Optical Networking Symposium Best Paper Awards at the IEEE Globecom 2007 and IEEE Globecom 2008 conferences. He won the 2004 UC Davis Distinguished Graduate Mentoring Award. He also won the 2009 UC Davis College of Engineering Outstanding Senior Faculty Award.

To date, he has supervised to completion the PhD Dissertations of 45 students, and he is currently supervising approximately 20 PhD students and research scholars.

He is author of the textbook "Optical WDM Networks" published by Springer in January 2006.

He served a 5-year term as a Founding Member of the Board of Directors of IPLocks, Inc., a Silicon Valley startup company. He has served on the Technical Advisory Board of a number of startup companies in networking, most recently Teknovus (acquired by Broadcom), Intelligent Fiber Optic Systems, and LookAhead Decisions Inc. (LDI).

He is a Fellow of the IEEE.